Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 9 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«ДСЛІДЖЕННЯ АЛГОРИТМІВ ОБХОДУ МАСИВІВ»

Варіант 5

Виконав студент ІП-11 Гіжицький Даниїл Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірив Мартинова Оксана Петрівна

( прізвище, ім'я, по батькові)

**Мета** – дослідити алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

**Варіант 5**

**Завдання:**



**Розв’язок:**

**1 Постановка задачі**

**Результатом виконання задачі сума елементів розташованих нижче головної діагоналі**

**2 Побудова математичної моделі**

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім'я** | **Призначення** |
| Матриця | Двовимірний масив дійсних чисел | matr | Вхідні та вихідні дані |
| Розмірність маттриці | Ціле | n | Проміжні дані |
| Сума | Ціле | sum | Результат |

**Створимо функцію** sum\_diag яка приймає як аргументи матрицю та посилання на функцію знаходження суми елементів розміщених під головною діагоналлю

Для ітерації по головній діагоналі використовуєм matr[i][i], а для побічної matr[i][j], де i – лічильник циклу (i < n), а n – розмірність матриці, а j лічильник циклу (j < i),

**Розв’язання**

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Деталізуємо функцію sum\_diag

Крок 2. Деталізуємо основну програму

**Псевдокод**

*Крок 1*

*функція* sum\_diag *(int n, list matr )*

**Початок**

Ініціалізація double sum = 0

**Повторити** для i від 1 до n

**Повторити** для j від 0 до i

sum += matr[i][j]

**Кінець повторити**

**Кінець повторити**

Ввиведення sum

**Кінець**

*Крок 2*

*Основна програма*

**Початок**

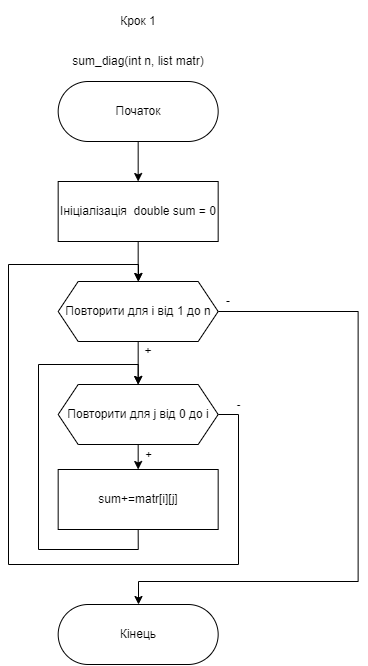
Введення matr

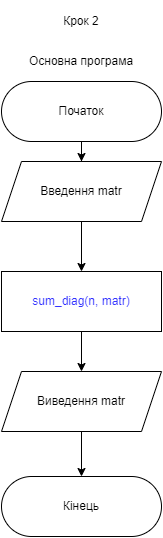
sum\_diag (n)

Виведення matr

**Кінець**

**Блок схема**

****

****

**Код програми (c++)**

#include <iostream>

#include<ctime>

using namespace std;

double\*\* rand\_matrix(int n) {

double\*\* mat = new double\*[n];

for (int i = 0; i < n; i++) {

double\* arr = new double[n];

for (int j = 0; j < n; j++)

arr[j] = rand() % 21 - 10;

mat[i] = arr;

}

return mat;

}

double sum\_diag(int n, double\*\* matr) {

double sum = 0;

for (int i = 1; i < n; i++)

{

for (int j = 0; j < i; j++)

{

sum += matr[i][j];

}

}

return sum;

}

void print\_matrix(int n, double\*\* matr) {

cout << "Matrix:\n";

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

cout << matr[i][j] << ' ';

}

cout << "\n";

}

cout << "\n";

}

int main()

{

srand(time(NULL));

int n;

cin >> n;

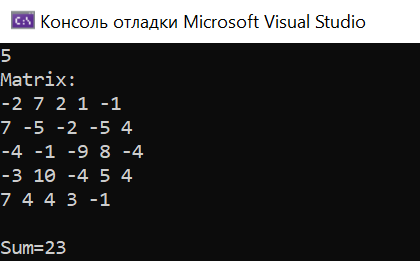
double\*\* matr = rand\_matrix(n);

print\_matrix(n, matr);

cout << "Sum=" << sum\_diag(n, matr);

}

**Тестування**



**Висновок**

Ми дослідили алгоритми обходу масивів та набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій і у результаті виконання програми отримали суму значень під головною діагоналлю матриці.